

## Parametrii subprogramelor

Tip de parametru		Folosire
Parametru actual		Apare în apelul funcției. Parametrii corespunzători pot fi atât valoare cât și referință
Parametru formal	Transmis prin Valoare (Parametru valoare formal)	Apare în header-ul funcției. Primește o copie a valorii păstrate în parametrul actual corespunzător
	Transmis prin Referință (Parametru referință formal)	Apare în header-ul funcției. Primește adresa parametrului actual corespunzător

Atunci când se **execută** o funcție, ea folosește **parametrii actuali** care i-au fost transmiși prin **apelul** său, ținând, însă, cont de natura parametrilor formali. Limbajul C++ acceptă două tipuri de parametri **formali**: **parametri valoare** și **parametri referință**.

### Parametrii valoare

- funcția primește o copie a valorii parametrului actual.
- deoarece parametrii valoare primesc copii ale parametrilor actuali, se pot folosi constante, variabile și expresii în apelul unei funcții.

Header: `void Functie2(double val, int contor);`

Apeluri corecte: `Functie2(y, i);`

`Functie2(9.81, i);`

`Functie2(4.9*sqrt(y), i);`

`Functie2(y, 3);`

- Numărul parametrilor actuali din apelul unei funcții trebuie să fie egal cu numărul parametrilor formali din header-ul funcției. De asemenea, tipurile datelor trebuie să corespundă. Dacă tipurile de dată nu se potrivesc, compilatorul încearcă să aplice operațiile de conversii implicite de tip.

- Când funcția se încheie, conținutul parametrilor valoare este șters, la fel cum se întâmplă în cazul variabilelor locale. Diferența dintre parametrii valoare și variabilele locale este că cele din urmă sunt nedefinite la startul funcției, în timp ce primele sunt inițializate automat cu valorile corespunzătoare ale parametrilor actuali.

**Parametrii referință** sunt cei pentru care:

- tipul de dată este însoțit, în lista parametrilor formali, de semnul &. Funcția primește adresa de memorie a parametrului actual.
- sunt parametri de intrare-ieșire, deoarece sunt folosiți pentru a transmite date dinspre blocul apelant către blocul apelat, dar și invers, la terminarea execuției, blocul apelat transmite date către blocul apelant.
- numele parametrului actual și cel al parametrului formal devin *sinonime* pentru aceeași variabilă. Ceea ce va lăsa funcția în acea locație va fi regăsit de funcția apelant.
- către parametrii referință pot fi transmise doar nume de variabile, nu și de constante sau expresii

Header:    void Functie2(double val, int& contor);

Apeluri corecte: Functie2(y, i);

Functie2(9.81, i);

            Functie2(4.9\*sqrt(y), i);

Apel incorect: Functie2(y, 3);

- O altă diferență importantă între parametrii valoare și cei referință apare la verificarea potrivirii tipurilor de dată între parametrii actuali și cei formali. Dacă în primul caz se realizează conversii implicite, pentru valorile referință lucrurile sunt diferite.